

SCHEDULING OF FLEXIBLE MANUFACTURING SYSTEM USING INTELLIGENT HEURISTIC SEARCH ALGORITHM (IHSA*)

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR THE DEGREE IN
BACHELOR OF TECHNOLOGY
IN
MECHANICAL ENGINEERING**



**BY
PRADYUMNA KUMAR GIDHI
ROLL NO. 108ME046
DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA
2008-1012**

SCHEDULING OF FLEXIBLE MANUFACTURING SYSTEM USING INTELLIGENT HEURISTIC SEARCH ALGORITHM (IHSA*)

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR THE DEGREE IN

BACHELOR OF TECHNOLOGY

IN

MECHANICAL ENGINEERING

BY

PRADYUMNA KUMAR GIDHI

ROLL NO. 108ME046

UNDER THE SUPERVISION OF

PROF. B. B. BISWAL



DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

2008-2012

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA



CERTIFICATE

This is to certify that the thesis entitled, “*Scheduling of Flexible Manufacturing Systems using Intelligent heuristic search algorithm (IHSA*)*” submitted by **Pradyumna Kumar Gidhi** in partial fulfilment of the requirements for the award of Bachelor of Technology Degree in Mechanical Engineering at National Institute of Technology, Rourkela (Deemed University), is an authentic work carried out by him under my supervision.

To the best of my knowledge the matter embodied in the thesis has not been submitted to any University/Institute for the award of any Degree or Diploma.

Date:

Prof. B.B.Biswal
Department of Mechanical Engineering
National Institute of Technology
Rourkela-769008



ACKNOWLEDGEMENT

I avail this opportunity to extent my hearty indebtedness to my guide **Prof. B.B. Biswal**, Mechanical Engineering Department, for their valuable guidance, constant encouragement and kind help at different stages for the execution of this dissertation work.

I also express my sincere gratitude to **Prof. K.P. Maity**, Head of the Department, Mechanical Engineering, for providing valuable departmental facilities and **Prof. C.K. Biswas** and the **Production Panel**, for constantly evaluating me and for providing useful suggestions.

SUBMITTED BY-

PRADYUMNA KUMAR GIDHI

ROLL NO. 108ME046

MECHANICAL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA-769008

CONTENTS

SL NO.	TOPIC	PAGE NUMBER
1	CHAPTER 1 INTRODUCTION	1-3
2	CHAPTER 2 LITERATURE RIVIEW	4-7
3	CHAPTER 3 METHODOLOGY	8-16
4	CHAPTER 4 ILLUSTRATIVE PROBLEM	17-23
5	CHAPTER 5 DISCUSSION	24-25
6	CHAPTER 6 CONCLUSION	26-27
7	REFERENCES	28-29

ABSTRACT

The complete scheduling of FMS includes two independent processes: sequencing of jobs and scheduling those prioritized jobs. In a flow shop or a Progressive type FMS, scheduling problem involves sequencing of 'n' jobs on 'm' machines with minimum makespan. Intelligent heuristic search algorithm (IHSA) is used in this paper, which ensure to find an optimal solution for flow-shop problem involving arbitrary number of machines and jobs provided the job sequence is same on each machine. The initial version of IHSA* is based on the A* algorithm. The final version of IHSA* is the modification of the initial IHSA*. There are three modifications: first modification concerned with the selection of an admissible heuristic function, second modification concerned with the procedure which determine heuristic estimate as the search progresses and the third modification concerned with the searching of multiple optimal solution, if they exist. Both version of the IHSA* are presented in this paper with an example which illustrates the use of both.*

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

The continuous increase in market competitiveness has increased the standards for product quality and performance while reduced the product development time and unit cost. Global competitiveness, fluctuating market requirements and modern lifestyle trends have put up tremendous challenge to manufacturing industries. In the current business scenario the competitiveness of any manufacturing industry is determined by its ability to respond quickly to the rapidly changing market and to produce high quality products at low costs. Other than cost, flexibility, quality, efficient delivery and customer satisfaction are also competitive factor and these draws equal attention as that of cost. These capabilities can be achieved by automation, robotics and other innovative concepts such as just-in-time (JIT), Production planning and control (PPC), etc. Flexible manufacturing is a concept that allows manufacturing systems to be built under high customized production requirements. Flexible manufacturing system provides agility to manufacturers. The agile manufacturer can be defined as the one who is fastest to the market, operates with the lowest total cost and has the greatest ability to delight its customers.

Flexible Manufacturing System (FMS) is an automated manufacturing system which consists of group of automated machine tools, interconnected with an automated material handling and storage system and controlled by computer to produce products according to the right schedule. FMS is called flexible because of its capability to process a variety of part style and quick response to changing demand patterns. There are various types of flexibility namely mix flexibility, volume flexibility, manufacturing flexibility and delivery flexibility.

There are various types of FMS:

i) *According to level of flexibility*

- a. Sequential FMS
- b. Random FMS
- c. Dedicated FMS
- d. Engineered FMS
- e. Modular FMS

ii) *According to types of layout*

- a. Progressive or line type FMS
- b. Loop type FMS

- c. Ladder type FMS
- d. Open field type FMS
- e. Robot centred type FMS

FMS is highly suitable for mid variety and mid volume part type production. It combines the productivity of transfer line or flow line and the flexibility of job shop. Proper sequencing and scheduling is required for high productivity. Sequencing is defined by Ashour [15] as being ‘concerned with the arrangements and permutations in which a set of jobs under consideration are performed on all machines.’ Baker [16] states “scheduling is the allocation of resources over time to perform a collection of tasks.”

This paper focuses on determining an optimal job sequence having minimum makespan using Intelligent heuristic search algorithm (IHSA*). Consider a progressive type FMS, as shown in Fig 1, where scheduling problem consisting of ‘m’ machine and ‘n’ jobs determines the job sequence on each machine with minimum makespan. Makespan is defined as the total time length starting from the first operation of the first job to the finishing of the last operation of the last job. When each job is processed on the machine 1,2,...,m, in that order then the number of possible schedule is $(n!)^m$. In progressive type FMS workstation, material handling and storage systems are arranged in a line. Here part or job progress from one workstation to the next in a well defined sequence with no back flow, very similar to transfer line or flow shop. IHSA* provides a procedure to find an optimal solution with arbitrary number of machine and jobs provided job sequence is same on each machine.

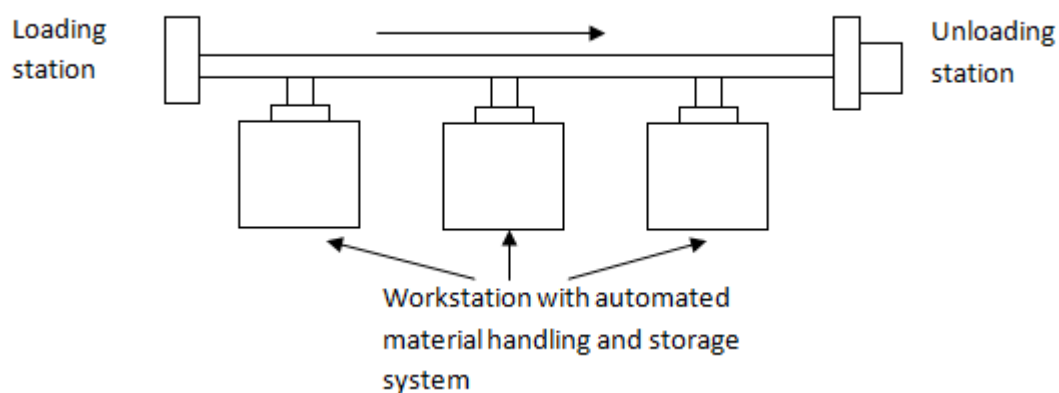


Fig 1: Progressive type FMS

CHAPTER 2

LITERATURE REVIEW

2. LITERATURE REVIEW

Scheduling of FMS is an ongoing research topic. The high investment and the high potential of FMS because of its adaptive nature, attracts many researcher. The performance of a Flexible manufacturing system (FMS) is highly depends on the selection of the right scheduling policy. Hence, there are many approaches and procedures have been developed for scheduling FMS and still the research is going on. There are many heuristic algorithms have been developed to solve different scheduling problems. All these algorithms aim to find an optimal solution or a near optimal solution efficiently.

Johnson [1] had given a procedure for finding an optimal solution with 2 machines or 3 machines using simple decision rules (FIFO, SPT, etc.).

Winley [2] developed an intelligent heuristic search algorithm for solving flow shop problem with two or three machine and arbitrary number of jobs. Then he modified its initial version of the algorithm to reduce the backtracking and to improve performance of the algorithm. The first modification is concerns with the choice of the best heuristic function and the second modification concerns with the procedure for determining the heuristic estimates at the nodes on the search path as the search progress. After the algorithm processed it gives the optimal job sequence having the minimum makespan.

Fan and Winley [3], developed a new intelligent heuristic search algorithm which guarantees to find an optimal solution for flow-shop problems with arbitrary number of jobs and arbitrary number of machine. There are three modification made to the initial version of IHSA*. The first modification concerns with the choice of an admissible heuristic function. The second modification concerns with the procedure for calculation of heuristic estimates as the search for the optimal solution progresses. The third modification concerns with the finding of multiple optimal solution when they exist.

Y. K. Kim, J. Y. Kim and K. S. Shi [4] , proposed a symbiotic evolutionary algorithm, named asymmetric multileveled symbiotic evolutionary algorithm(AMSEA).The scheduling of FMS problem consisting of loading, routing and sequencing sub-problems are interrelated to each other. AMSEA has the strength to simultaneously solve sub-problems for loading, routing and sequencing and can easily handle a variety of FMS flexibilities.

Baker [5], examined a basic scheduling model which involved trade-off between efficiency and due date performance. Heuristic solution procedures are examined for scheduling jobs on a single machine to minimize the maximum lateness in the presence of setup times between different job families, followed by designing of a new heuristic procedure for computational evaluation. Lateness is defined as the difference between the job completion time and the due date.

Angsana and Passino [6], developed an adaptive fuzzy controller at each machine which can automatically synthesize itself according to machine parameter variation. When parts arrive at a machine for processing from another machine, they are put in a buffer (queue) where they are held before they are processed. The adaptive fuzzy controller use the machine's buffer level to decide which part type to process next, hence the overall controller for the FMS is physically distributed across the entire FMS with the local controller at each machine. It also shows that a distributed fuzzy controller (DFC) can be generated from the adaptive fuzzy controller. DFC can automatically synthesize itself and lower the maximum buffer level more effectively than conventional scheduler. It improves the performance of the FMS but, in certain case, it is, in general, hard to choose certain parameters in the DFC and its performance is not always as good as conventional schedulers.

Jeng and Chen [7], proposed a new heuristic search method based on an analytic theory of the Petri net state equations for scheduling flexible manufacturing systems. The goal of this method is to minimize makespan and also to reduce the searched state space.

Laha and Sapkal [8], developed a constructive heuristic for NP hard problem of no wait flow shop scheduling problem. It is based on the assumption that the priority of a job in the sequence is given by the sum of its processing times on the bottleneck machines for selecting the initial job sequence of the jobs.

Priore, Fuente, Pino and Puente [10], describe a case-based reasoning (CBR) approach, which analyses the system's previous performance, and acquires scheduling knowledge to determine the optimal dispatching rule. Genetic algorithm has been used in this approach. However one of the major drawbacks of this approach is that it needs a large number of simulations to generate test training examples, but it only needed once.

Haq, Karthikeyan and Dinesh [11], described a scheduling problem of a job shop with AGVs by employing a heuristic, namely Giffler and Thompson algorithm [12], which is

a combination of evolutionary Genetic algorithm and Simulated Annealing algorithm. The main objective is to minimise makespan with sub-objective to minimise distance travelled and number of backtrackings by AGV.

Mati, Rezg and Xie [13], developed a taboo search heuristic which concerns with the allocation of resources such that the jobs are completed with a minimum makespan and deadlocks are avoided.

There are also other approaches and methods which involve multi-objective scheduling or dynamic scheduling. FMS consists of various components like workstation, material handling equipments, storage system etc. hence, it is necessary to schedule these component also in order to increase productivity of FMS. Some other approaches are integer linear programming [9]; evolutionary algorithm like generic algorithm, ant colony optimization method and particle swarm optimization. Artificial intelligence optimization methods include metahuristic approach such as simulated annealing, Tabu search.

CHAPTER 3

METHODOLOGY

3. METHODOLOGY

The methodology used here is the modified IHSA* [3]. The initial version of IHSA* is based on search and learning A* algorithm [14]. At the initial IHSA* computes estimate for the total time required to complete all the jobs on all the machine, using different methods; assuming each job is placed first in the job sequence. It is observed that for m machines there are m different methods which should be considered. The method with the smallest estimate is considered as the value of the heuristic function and it determines the job which should be placed first in the job sequence at the beginning of the search if that method is used. Heuristic function is said to be admissible if the value of the heuristic function does not exceeds the minimum makespan value for the problem then in such case IHSA* guarantee to find an optimal solution provided the sequence of the job is same on each machine. This method can be implemented in progressive type FMS involving arbitrary number of machine and job provided job sequences is same on each machine. If there are n number of jobs and m number of machine then total number of possible schedule is $(n!)^m$, when job sequence is not same on each machine and $(n!)$ when the job sequence is same on each machine.

There are three significant modification made to the initial IHSA*. The first modification concerns with the choice of an admissible heuristic function at the beginning of the search. The second modification concerns with the calculation of heuristic estimates as the search progresses and the third modification concerns with the finding of multiple optimal solution when they exist.

3.1 THE STATE TRANSITION PROCESS

At the beginning of the search all the $m*n$ operations are only one of the 3 states: the not scheduled state; the in-progress state; or the finished state. The operation which is selected is from those in the not scheduled state. Operations that are not in finished state are referred as incomplete. The state transition occurs when one or more of the operations move from the in-progress state to the finished state.

The procedure associated with the state transition process is described by the IHSA* and it can be illustrated graphically using search path diagrams.

3.2 SEARCH PATH DIAGRAMS

A search path diagram contains nodes drawn at each state level and one of the nodes among these is connect to the next state level. Each node contains cells equal to the number of machines used. When a state transition occurs that is when one or more of the operation move from in-progress state to the finished state then one of the nodes having the minimum heuristic estimate is expanded and is connected to the next state level. Each node at the state level represents one of the different ways of starting operations that are in the not schedule state on the available machine. The cell j in each node is labelled with J_i and P_{ij} , which is either in not scheduled state or in progress, where J_i is the job number, P_{ij} is the time required to complete job J_i on machine M_j . A blank cell represents that machine is idle at this time.

At each node heuristic estimate is calculated using procedure used in step 2 of the algorithm explain below. The heuristic estimate calculated is the estimate of the time needed to complete the operation at the node as well as all of the other operations that are not in the finished state. The node which having the minimum heuristic estimate among all of the estimates for the nodes at that state level is selected for expansion. The value of $f = h + k$ is calculated near the selected node, where the edge cost (k) represents the time that has passed since the preceding state transition occurred. A comparison is made between f and h' where h' is the minimum heuristic estimate at the preceding state level. Based on that comparison the search path either backtracks to the node expanded at the preceding state level or moves forward to the next state level. If the backtracking occurs then the value of h' is changed to the current value of f and the search move back to that previously expanded node and if the path move forward then the value of edge cost (k) is recorded and it is the smallest value among all of the P_{ij} values in the cells

At the state level 0 there are n root nodes corresponding to the number of jobs. The search stops at the node where $f = h = 0$, called the terminal node. The final search path diagram shows the optimal solution and traces a path from one of the root nodes to the terminal node and the value of h or f associated with that root node gives the value of minimum makespan. The optimal job sequence can be obtained by recording the completed operation along the final search path.

3.3 NOTATIONS USED

Consider a flow shop problem involving 3 machine and n jobs J_1, J_2, \dots, J_n as shown in table 1. O_{ij} is the operation performed on the job J_i by machine M_j . Total number of operations are $3n$. For job J_i the processing times a_i, b_i and c_i denote the time required to perform the operations O_{i1}, O_{i2} and O_{i3} respectively and are assumed to be non negative. If O_{ij} has started but has not been completed then P_{ij} denotes the extra time required to complete O_{ij} and at the time when O_{ij} starts P_{ij} is one of the value among a_i, b_i or c_i . Fig 2 shows the processing of the job sequence Φ_{st} on n jobs on the three machines. The sequence $\Phi_{st} = \{J_s, \dots, J_t\}$, where J_s is the job which is scheduled first and J_t is the last schedule job. $T(\Phi_{st})$ is the makespan for the job sequence Φ_{st} and $S(\Phi_{st})$ is the time at which all jobs in Φ_{st} are completed on machine M_2 .

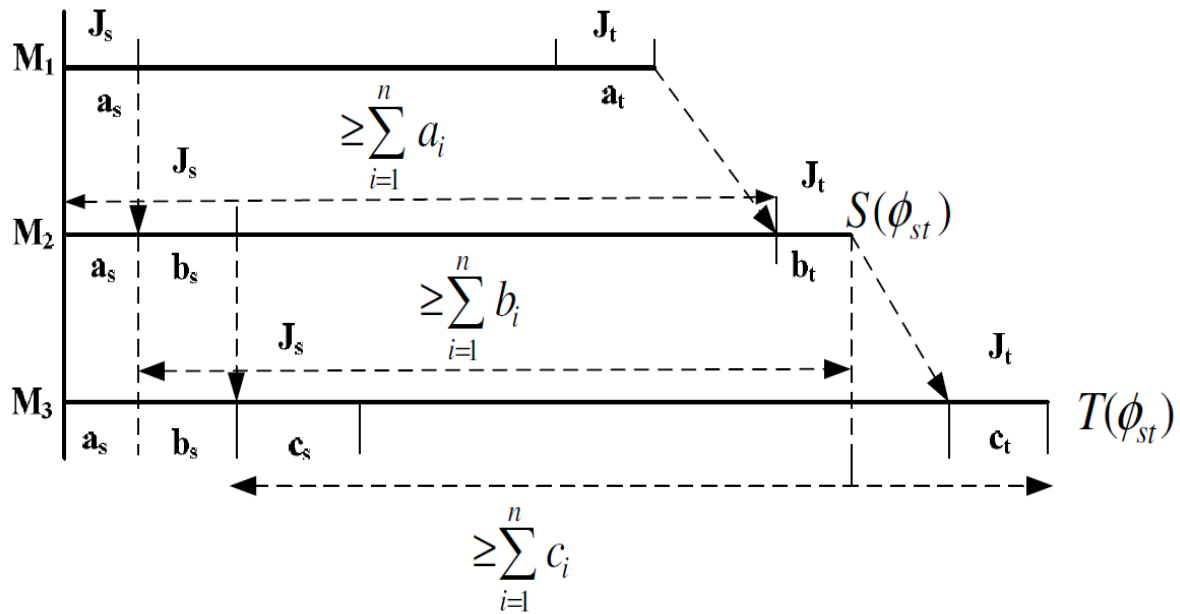


Fig 2: Processing of the job sequence Φ_{st} on different machines [3]

From Fig.2 it is observed that the total time required for completing the all the n jobs in all the machine when job J_i is the first job in the job sequence is given by

$$F_3 = \min [a_1 + b_1, a_2 + b_2, \dots, a_s + b_s, \dots, a_n + b_n] + \sum_{i=1}^n (c_i) \quad \dots \dots \dots (1)$$

If $\min [a_1 + b_1, a_2 + b_2, \dots, a_s + b_s, \dots, a_n + b_n] = a_s + b_s$ then job J_s is scheduled first in the job sequence.

3.4 THE INITIAL VERSION OF IHSA*

To find out the optimal job sequence with minimum makespan the following steps are followed:

Step 1: At state level 0, calculate F_3 from (1) and expand the node which is identified by it to the state level 1. Break the tie randomly, if more than one node is identified.

Step 2: Calculate the heuristic estimate using procedure 1 which is described below. If at the current state level, the heuristic estimate of one of the node has been updated by backtracking then use the updated value as the heuristic estimate for that node.

Step 3: At the current state level, select the node having the smallest heuristic estimate. Break the tie randomly if necessary.

Step 4: Calculate $f = h + k$, at the selected node, where h is the smallest heuristic estimate found in step 3 and edge cost(k) is the time that has elapsed since the preceding state transition occurred.

Step 5: Check if $f > h'$, where h' is the minimum heuristic estimate calculated at the preceding state level.

If yes then backtrack to the preceding state level and update the value of h' with the current value of f and repeat step 4 at that node.

If no then proceed to the next state level and repeat from step 2.

Step 6: If $f = 0$ and $h = 0$, then stop.

3.4.1 PROCEDURE 1:

It is used to calculate a heuristic estimate for each node in step 2

(a) If cell 1 is labelled with J_i then the heuristic estimate h for the node is based on the operation in cell 1 and is given by,

$$h = \begin{cases} a_i + b_i + C_1; \text{ for } O_{i1} \text{ is in the not scheduled state} \\ P_{i1} + b_i + c_i + C_1; \text{ for } O_{i1} \text{ is in the in-progress state} \end{cases} \dots\dots\dots(2)$$

Where C_1 is the sum of the values of c_k for all value of k such that O_{k1} is in the not scheduled state.

(b) If cell 1 is blank, and cell 2 is labelled with J_i then the heuristic estimate h for the node is based on the operation in cell 2 and is given by,

$$h = \begin{cases} b_i + C_2; \text{ for } O_{i2} \text{ is in the not scheduled state} \\ P_{i2} + c_i + C_2; \text{ for } O_{i2} \text{ is in the in-progress state} \end{cases} \dots\dots\dots(3)$$

Where C_2 is the sum of the values of c_k for all value of k such that O_{k2} is in the not scheduled state.

(c) If cell 1 and cell 2 are blank, and cell 3 is labelled with J_i then the heuristic estimate h for the node is based on the operation in cell 3 and is given by,

$$h = \begin{cases} C_3; \text{ for } O_{i3} \text{ is in the not scheduled state} \\ P_{i3} + C_3; \text{ for } O_{i3} \text{ is in the in-progress state} \end{cases} \dots\dots\dots(4)$$

Where C_3 is the sum of the values of c_k for all value of k such that O_{k3} is in the not scheduled state.

3.5 MODIFICATION TO THE INITIAL IHSA*

Winley and Fan [3] modified the initial version of IHSA* and develop six heuristic functions for three machine and n job problem. Among these 6 heuristic functions three are admissible as having initial values which are closest to the minimum makespan. The three admissible heuristic functions are as shown below:

$$\left. \begin{aligned} F_1 &= \min [b_1 + c_1, b_2 + c_2, \dots, b_n + c_n] + \sum_{i=1}^n (a_i) \\ F_2 &= \min [a_1 + u_1, a_2 + u_2, \dots, a_n + u_n] + \sum_{i=1}^n (b_i) \\ F_3 &= \min [a_1 + b_1, a_2 + b_2, \dots, a_n + b_n] + \sum_{i=1}^n (c_i) \end{aligned} \right] \dots\dots\dots(5)$$

Where, $u_1 = \min [c_2, c_3, \dots, c_n]$

$u_k = \min [c_1, c_2, c_3, \dots, c_{k-1}, c_{k+1}, c_n], \text{ for } 2 \leq k \leq n-1$

$u_n = \min [c_1, c_2, c_3, \dots, c_{n-1}]$

The heuristic function having the largest value is selected as the admissible heuristic function in the first step. It ensures that the search begins with an estimate an estimate of minimum makespan which is not greater than but closer to it.

For arbitrary number of machine, the best heuristic function is one of among $F_1, F_2, F_3, \dots, F_m$ having the largest value where,

$$F_j = \left[\begin{aligned} &\min [\sum_{i=2}^m (t_{1,i}), \sum_{i=2}^m (t_{2,i}), \dots, \sum_{i=2}^m (t_{n,i})] + \sum_{i=1}^n (t_{i,1}), \text{ for } j=1 \\ &\min [\sum_{i=1}^{j-1} (t_{1,i} + u_{j,1}), \sum_{i=1}^{j-1} (t_{2,i} + u_{j,2}), \dots, \sum_{i=1}^{j-1} (t_{n,i} + u_{j,n})] \\ &\quad + \sum_{i=1}^n (t_{i,j}), \text{ for } 2 \leq j \leq m \end{aligned} \right] \dots\dots\dots(6)$$

Where, for $2 \leq j \leq m-1$,

$$u_{j,k} = \left[\begin{aligned} &\min [\sum_{i=j+1}^m (t_{2,i}), \sum_{i=j+1}^m (t_{3,i}), \dots, \sum_{i=j+1}^m (t_{n,i})], \text{ for } k=1 \\ &\min [\sum_{i=j+1}^m (t_{1,i}), \sum_{i=j+1}^m (t_{2,i}), \dots, \sum_{i=j+1}^m (t_{k-1,i}), \\ &\quad \sum_{i=j+1}^m (t_{k+1,i}), \dots, \sum_{i=j+1}^m (t_{n,i})], \text{ for } 2 \leq k \leq n-1 \\ &\min [\sum_{i=j+1}^m (t_{1,i}), \sum_{i=j+1}^m (t_{2,i}), \dots, \sum_{i=j+1}^m (t_{n-1,i})], \text{ for } k=n \end{aligned} \right] \dots\dots\dots(7)$$

$u_{m,k} = 0, \text{ for } k=1, 2, 3, \dots, n.$

Where $t_{i,j}$ is the time required to perform $O_{i,j}$.

The procedure for calculating heuristic estimates is also modified. It calculate the heuristic estimate h_1 , h_2 , and h_3 (considering 3 machine and n job problem) at each node for cells 1,2, and 3 respectively. The maximum of h_1 , h_2 and h_3 is considered as the heuristic estimate (h) for the node.

3.6 THE FINAL VERSION OF IHSA*

Step 1: At state level 0, from (5) choose the admissible heuristic function among F_1 , F_2 and F_3 which has the largest value and if the number of machine $m > 3$ then from (6) choose the admissible heuristic function. Expand the node which is identified by the chosen admissible function. Break the tie randomly, if necessary.

Step 2: At the current state level, at each node use procedure 2 (which is mentioned below) to calculate h_1 , h_2 and h_3 and use $\max [h_1, h_2, h_3]$ as the heuristic estimate for that node. If the heuristic estimate of one of the nodes has been updated by backtracking use updated value as the heuristic estimate for that node and proceed to step 3. If $m > 3$ then procedure 2 is modified to incorporate admissible heuristic function F_j and is used to calculate h_1, h_2, \dots, h_m . The $\max [h_1, h_2, \dots, h_m]$ is used as the heuristic estimate for the node.

Step 3: At the current state level select the node with the smallest heuristic estimate for expansion. Break the tie randomly, if necessary.

Step 4: Calculate $f = h + k$, at the selected node, where h is the smallest heuristic estimate found in step 3 and edge cost(k) is the time that has elapsed since the preceding state transition occurred.

Step 5: Check if $f > h'$, where h' is the minimum heuristic estimate calculated at the preceding state level.

If yes then backtrack to the preceding state level and update the value of h' with the current value of f and repeat step 4 at that node.

If no then proceed to the next state level and repeat from *step 2*.

Step 6: if $f = 0$ and $h = 0$ then an optimal solution has been found. If along the path representing the optimal solution there is a node which was selected for expansion by breaking ties randomly among nodes at the same state level with the same minimum heuristic estimate then return to the that state level and repeat form step 2 ignoring any node that was

selected previously for expansion as a result of breaking ties. If any of the values of h at root nodes is less than or equal to the minimum makespan then return to state level 0 and repeat from *step 2* ignoring root nodes that lead to a previous optimal solution. Otherwise, stop.

Procedure 2

It is used to calculate a heuristic estimate for each cell at a node, in step 2.

(a) For cell 1: If cell is blank then $h_1=0$

Otherwise, h_1 is given by (2).

(b) For cell 2: If cell is blank then $h_2=0$

Otherwise, h_2 is given by (3).

(c) For cell 3: If cell is blank then $h_3=0$.

Otherwise, h_3 is given by (4).

$$\left. \begin{array}{l} \text{(a) For cell 1: If cell is blank then } h_1=0 \\ \text{Otherwise, } h_1 \text{ is given by (2).} \\ \text{(b) For cell 2: If cell is blank then } h_2=0 \\ \text{Otherwise, } h_2 \text{ is given by (3).} \\ \text{(c) For cell 3: If cell is blank then } h_3=0. \\ \text{Otherwise, } h_3 \text{ is given by (4).} \end{array} \right\} \dots\dots\dots(8)$$

CHAPTER 4

ILLUSTRATIVE PROBLEM

4. PROBLEM 1

Consider a flow shop or progressive type FMS involving 3 jobs and 3 machines. The processing times are shown in table below:

Jobs / Machines	M₁	M₂	M₃
J₁	4	3	5
J₂	2	4	8
J₃	5	3	6

This problem is solved using both initial IHSA* and the modified IHSA*.

Using initial IHSA*, from (1)

$$\begin{aligned}
 F_3 &= \min [a_1 + b_1, a_2 + b_2, \dots, a_s + b_s, \dots, a_n + b_n] + \sum_{i=1}^n (c_i) \\
 &= a_2 + b_2 + \sum_{i=1}^n (c_i) \\
 &= 6 + 19 \\
 &= 25
 \end{aligned}$$

The search path diagram is shown in Fig 3, Fig 4 and Fig 5. Three search path diagrams are drawn to properly illustrate the working of the algorithm.

After expanding the nodes it is found that minimum makespan is 25 and the optimal job sequence is J₂ J₃ J₁.

Using modified IHSA*, from (5)

$$F_1 = 19, F_2 = 17 \text{ and } F_3 = 25$$

Among these F₃ has a value greater than other two; hence F₃ is the admissible heuristic function.

Expanding nodes we get two optimal job sequences, J₂ J₁ J₃ and J₂ J₃ J₁ with minimum makespan of 25. The search path diagram is shown in Fig 6 and Fig 7.

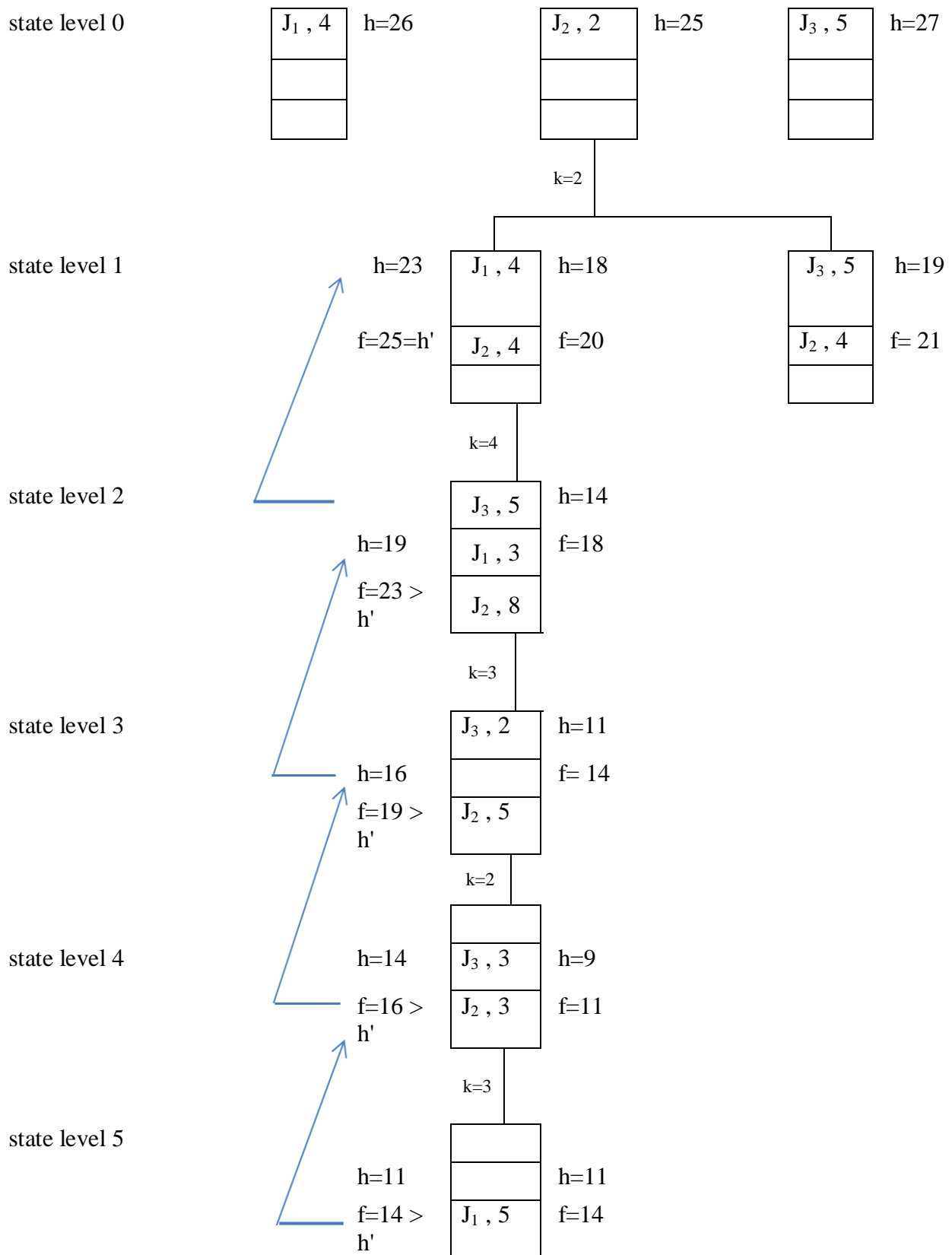


Fig 3: The first search path diagram for problem 1 using initial IHSA*

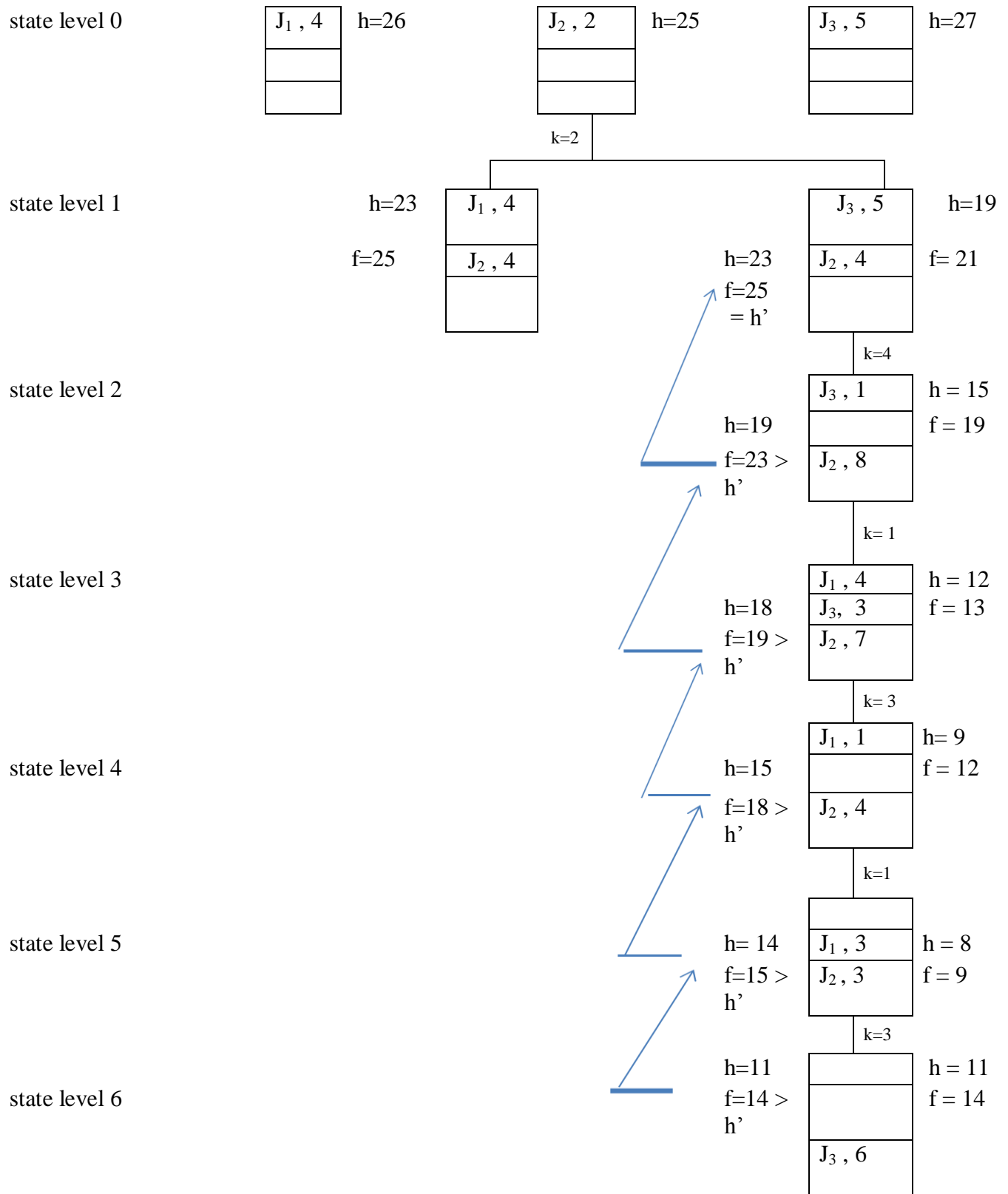


Fig 4: The second search path diagram for problem 1 using initial IHSA*

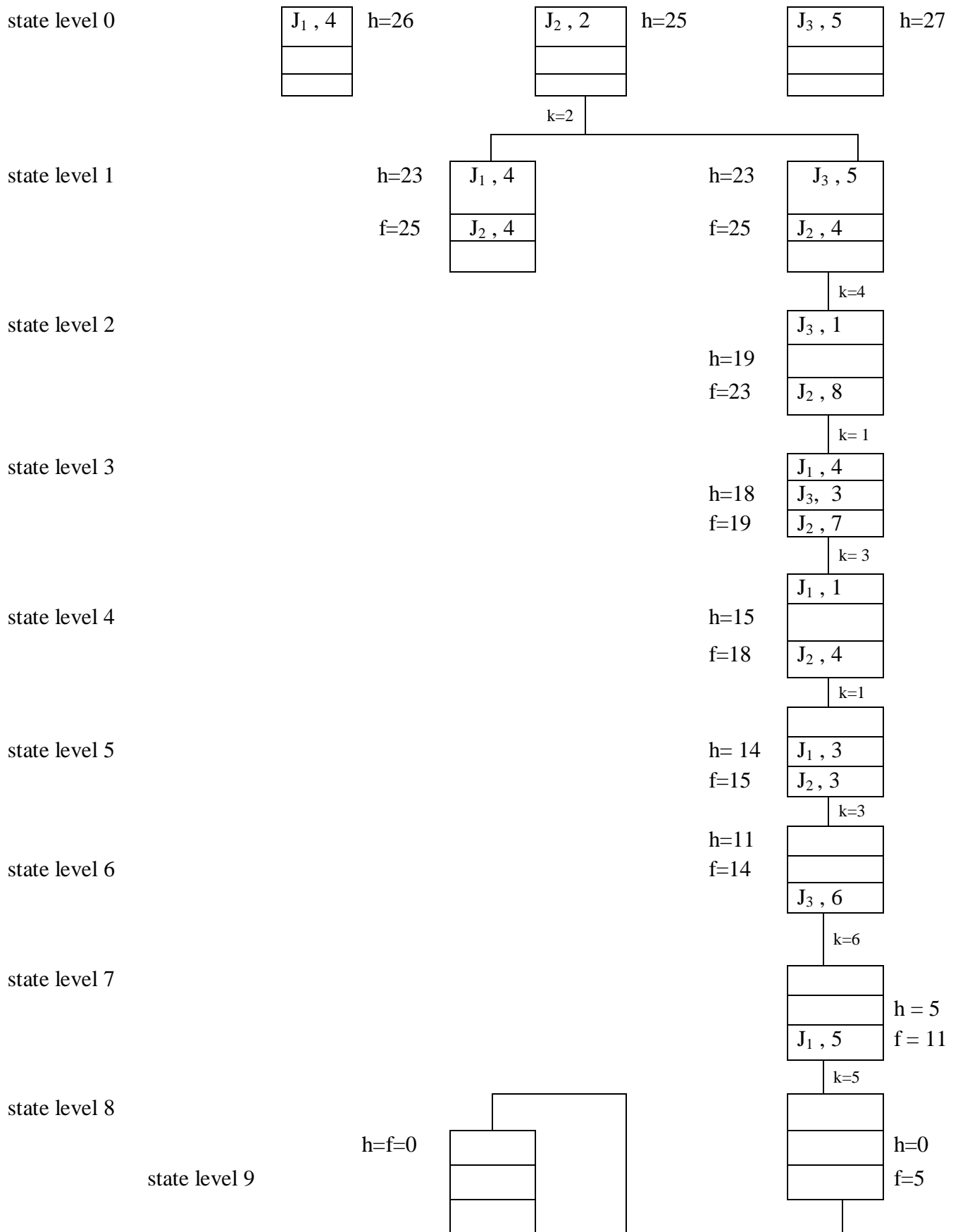


Fig 5: The final search path diagram for problem 1 using initial IHSA*

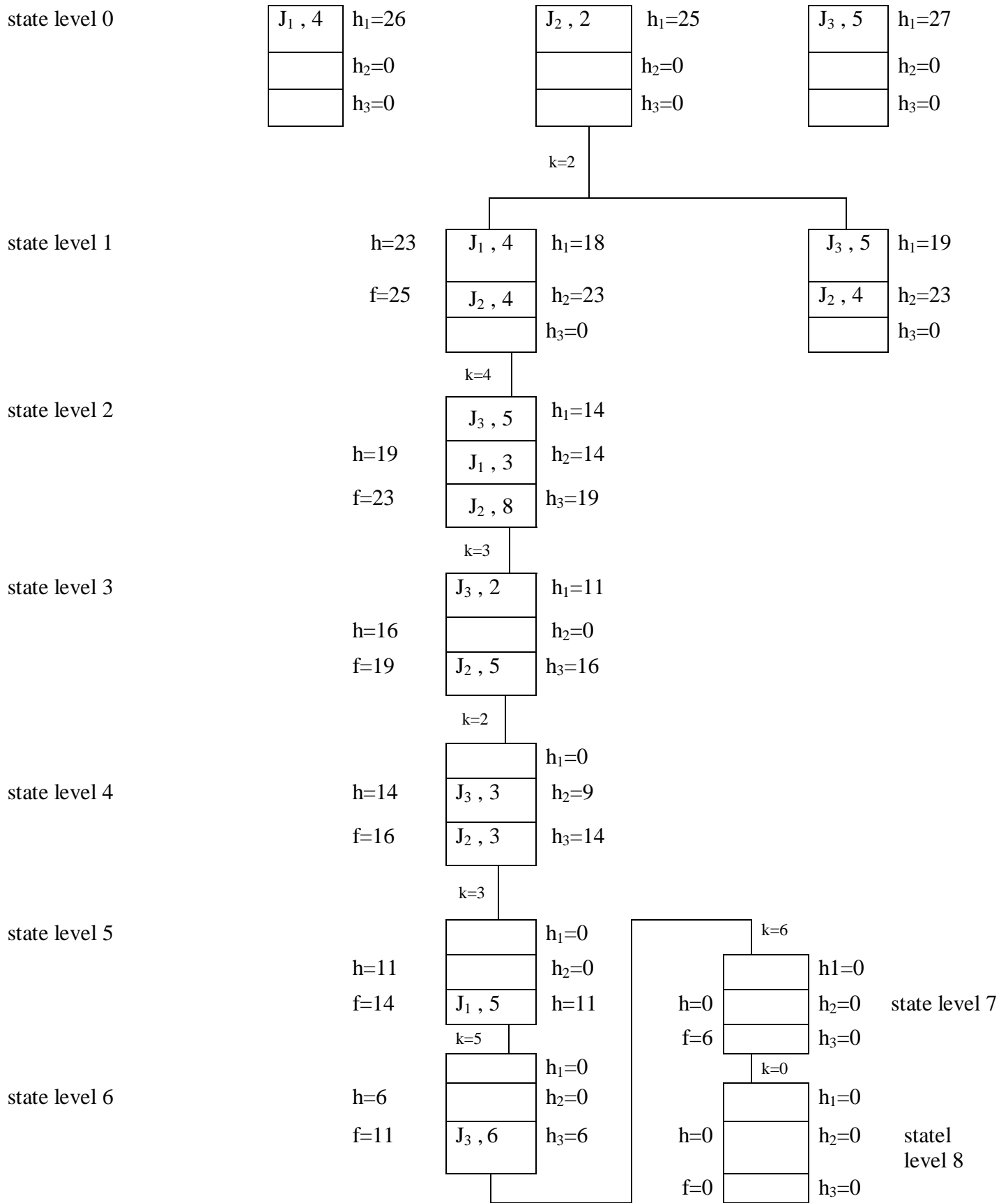


Fig 6: The search path diagram for problem 1 using modified IHSA*

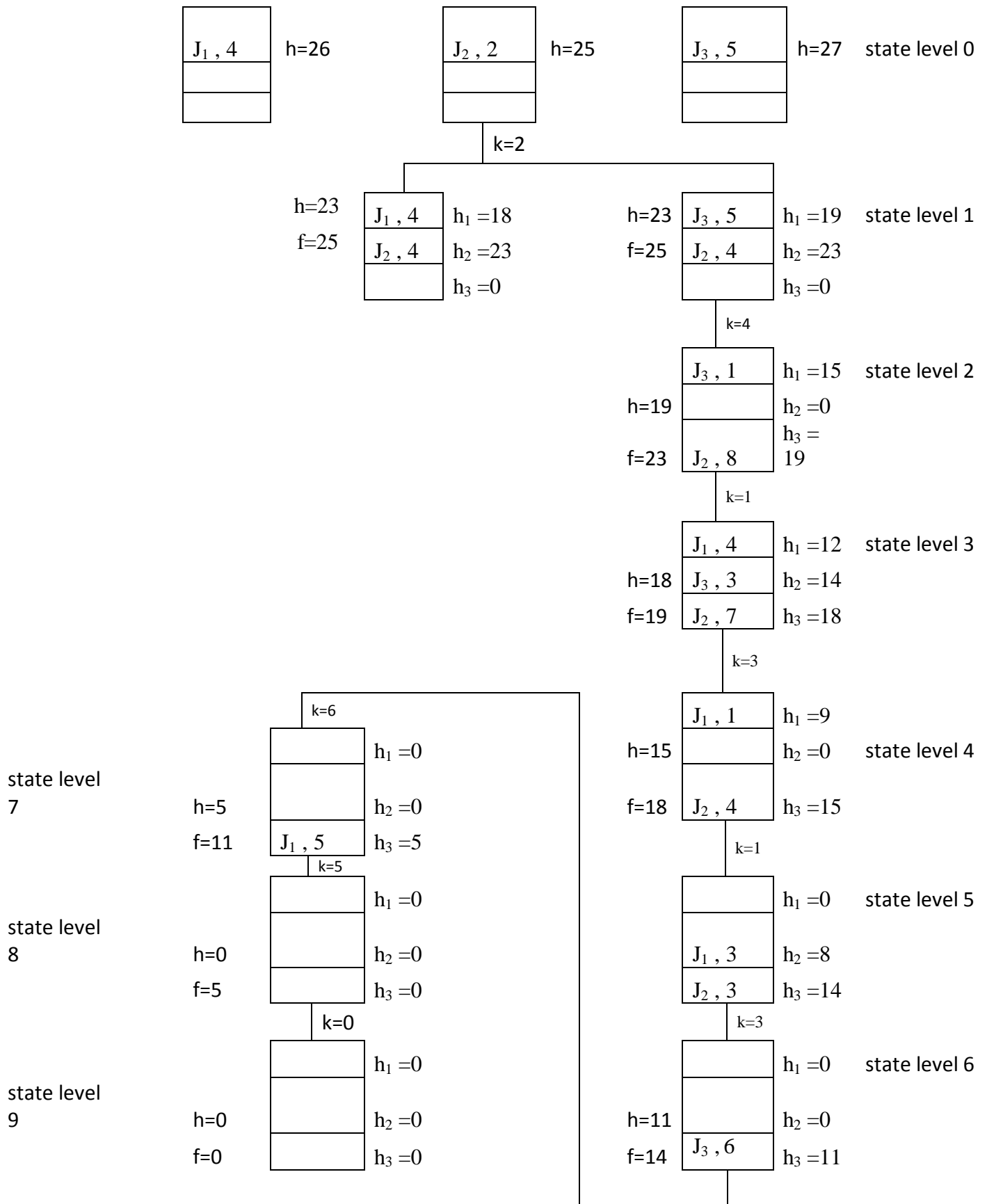


Fig 7: The search path diagram for problem 1 using modified IHSA*

CHAPTER 5

DISCUSSION

5. DISCUSSION

The above problem was solved by both initial and final version of IHSA*. The performance of both can be explained by the number of nodes expanded and the number of backtracking steps required. The optimal job sequence $J_2 J_3 J_1$ with minimum makespan of 25 is found out by expanding 20 nodes and 9 backtracking steps, using initial IHSA*. Fig 3, Fig 4 and Fig 5 shows the search path diagram for the given problem. The final version of IHSA* generate two optimal job sequences $J_2 J_1 J_3$ and $J_2 J_3 J_1$ with minimum makespan of 25. Total number of node expanded is 18 with no backtracking. The search path diagram is shown in Fig 6 and Fig 7. Two search paths are drawn to show the modification in the last step which ensures to find multiple optimal solution if they exist.

The procedure 1 and procedure 2, given for calculation of heuristic estimate are for heuristic function F_3 . However, the procedure can be modified to incorporate the other admissible heuristic function. The use of smallest of the largest estimate at the node reduces the likely occurrence of backtracking and improves the performance characteristic of the algorithm.

CHAPTER 6

CONCLUSION

6. CONCLUSION

For a flow shop scheduling problem consisting of m machines and n jobs, the total number of schedule in which job can be processed is given by $(n!)^m$ when the job sequence is not same. When same job sequence is used on each machine then the number of schedule is reduced to $n!$. For problem 1, the total number of schedule is $3! = 6$. The final version of IHSA* finds two optimal solutions with expanding less node and with 0 backtracking. However, when number of machine increases; the number of heuristic function increases accordingly and when number of jobs increases; the number of schedule increases, ultimately increasing machine and job increases the search space. At this point, the modified IHSA* provides better result than the initial IHSA* with less execution time.

REFERENCES

- [1] Johnson, S.M. "Optimal two and three state production schedules with setup time included," *Naval Research Logistics Quarterly*, 1: (1), 1954, 61-68.
- [2] Winley, G. "Flow Shop problem: A heuristic search algorithm," *AU J.T* 10 (1): 2006,11-12.
- [3] Fan, J.P.O., Winley, G.K. "A heuristic search algorithm for flow-shop scheduling," *Informatica* 32, 2008, 453-464.
- [4] Kim, Y.K., Kim, J.Y., Shin, K.S., "An asymmetric multileveled symbiotic evolutionary algorithm for integrated FMS scheduling," *J Intell Manuf* 18: 2007, 631-645
- [5] Baker, K.R. "Heuristic Procedures for Scheduling Job Families with setups and due dates".
- [6] Angsana, A., Passino, K.M. "Distributed Fuzzy control of Flexible Manufacturing Systems," *IEEE Transactions on Control System Technology*, Vol. 2, No. 4: 1994, 423-435.
- [7] Jeng, M.D., Chen, S.C. "A heuristic search approach using approximate solutions to Petri Net state equations for scheduling Flexible manufacturing systems," *The International Journal of Flexible Manufacturing Systems*, 10: 1998, 139-162.
- [8] Laha, D., Sapkal, S.U. "An efficient heuristic algorithm for m- machine no-wait flow shops," *IMECS*, Vol 1, 2011.
- [9] Pan, C.H. "A study of integer programming formulation for scheduling problems," *International Journal of System Scinece*, 28: (1), 1997, 33-41.
- [10] Priore, P., Fuente, D.D.L., Pino, R., Puente, J. " Learning based scheduling of Flexible Manufacturing System using case-based reasoning,"

- [11] Haq, A.N., Karthikeyan, T., Dinesh, M. "Scheduling decisions in FMS using heuristic approach," Int J Adv Manuf Technol, 22: 2003, 374-379.
- [12] Giffler, B., Thompson, G.L. "Algorithms for solving production scheduling problems," Int J Oper Res 8: 1960, 487-503.
- [13] Mati, Y., Rezg, N., Xie, X. "Geometric approach and Taboo Search for scheduling Flexible Manufacturing System," IEEE Transactions on Robotics and Automation, Vol 17, No. 6, 2001, 805-818.
- [14] Zamani, R., Shue, L.Y. "Solving project scheduling problems with a heuristic learning algorithm," Journal of the Operational Research Society, 49: (7), 1998, 709-716.
- [15] Ashour, S. "Sequencing Theory," Springer-Verlag, New York, 1972.
- [16] Baker, K.R. "Introduction to Sequencing and Scheduling," John Wiley and Sons, Inc., New York, 1974.
- [17] Shivanand, H.K., Benal, M.M., Koti, V. "Flexible Manufacturing System."